## Prerequisites:

If you don’t have anaconda/miniconda,

Install miniconda

<https://docs.conda.io/en/latest/miniconda.html>

Create environment with python = 3.7

conda create -n dl python=3.7

Install pytorch-cpu

conda install -c pytorch pytorch-cpu

If that doesn’t work try:

conda install -c conda-forge pytorch-cpu

Start coding :)

Stl\_10binary contains the data

The skeletal code is 3\_CNN\_classification\_exercise.py

Search for # \* (hash space asterisk) in the file

## Objective

Implement various layers to train a deep neural network. The network architecture to be implemented can be found in the lecture slide. Train the network for image classification and consider the following questions for the report.

## Implementation

The code provided to you consists of two training routines: first, for a two-class classification problem and secondly, for a ten-class classification problem. The two-class classification problem is setup by simply selecting two classes from the STL-10 dataset (ship and cat). The second ten-class classifier works with the entire dataset. This separation into two problems has been done as you will be running your networks on CPUs. The running times on CPUs are much longer as compared to GPUs, which are usually used for training such networks. Nevertheless, as you will see, several interesting insights can already be obtained by running experiments on smaller datasets.

### Network Architecture

The network architecture is to be implemented inside the ’CNN’ class. Add attributes/elements in the CNN class and implement the said layers. Modify the implementation of the ’forward’ function suitably. Create a new instance of the network and train it again. Do you observe a difference in performance for the two-class problem? How about for the ten-class problem? Play around with the number of layers in the networks and the number of filters in the different layers. Do you observe any trends in performance as the network architecture is varied? How does the training time depend on the network architecture? What is the best test accuracy that you could obtain for the 2-class and the 10-class classification problems?

### Training and testing

Discrepancy between training and test error Is there a large gap between the accuracies on the training and the test set in the 10-class problem and 2-class problem? If so, what could be the reasons for this behaviour? What is over-fitting? Do you know any measures to prevent over-fitting?

Network Architecture to be implemented

Input image → 3x3 convolutional layer with stride 1 and 32 output feature maps

→ activation layer

→ max-pooling layer with 2x2 spatial extent and stride 2

→ 3x3 convolutional layer with stride 1 and 32 output feature maps

→ activation layer

→ max-pooling layer with 2x2 spatial extent and stride 2

→ 3x3 convolutional layer with stride 1 and 32 output feature maps

→ activation layer

→ max-pooling layer with 2x2 spatial extent and stride 2

→ vectorize the feature maps

→ fully-connected layer with 64 output units

→ activation layer

→ fully-connected layer with 2/10 output units.